2018-12

Behavioral Learning in a Cognitive Neuromorphic Robot: An Integrative Approach

Wennekers, Thomas

10.1109/TNNLS.2018.2816518
IEEE Transactions on Neural Networks and Learning Systems
Institute of Electrical and Electronics Engineers

Behavioral Learning in a Cognitive Neuromorphic Robot: An Integrative Approach

Alexander D. Rast, *Member, IEEE,* Samantha V. Adams, Simon Davidson, Sergio Davies, Michael Hopkins, Andrew Rowley, Alan Barry Stokes, Thomas Wennekers, Steve Furber, *Fellow, IEEE,* and Angelo Cangelosi

***Abstract*— We present here a learning system using the iCub humanoid robot and the SpiNNaker neuromorphic chip to solve the real-world task of object-specific attention. Integrating spiking neural networks with robots introduces considerable complexity for questionable benefit if the objective is simply task performance. But, we suggest, in a cognitive robotics context, where the goal is understanding how to compute, such an approach may yield useful insights to neural architecture as well as learned behavior, especially if dedicated neural hardware is available. Recent advances in cognitive robotics and neuromorphic processing now make such systems possible. Using a scalable, structured, modular approach, we build a spiking neural network where the effects and impact of learning can be predicted and tested, and the network can be scaled or extended to new tasks automatically. We introduce several enhancements to a basic network and show how they can be used to direct performance toward behaviorally relevant goals. Results show that using a simple classical spike-timing-dependent plasticity (STDP) rule on selected connections, we can get the robot (and network) to progress from poor task-specific performance to good performance. Behaviorally relevant STDP appears to contribute strongly to positive learning: "do this" but less to negative learning: "don't do that." In addition, we observe that the effect of structural enhancements tends to be cumulative. The overall system suggests that it is by being able to exploit combinations of effects, rather than any one effect or property in isolation, that spiking networks can achieve compelling, task-relevant behavior.**

***Index Terms*— Cognitive, learning, multiscale, neuromorphic, robotics, spike-timing-dependent plasticity (STDP).**

Manuscript received June 27, 2016; revised October 4, 2017 and February 19, 2018; accepted March 5, 2018. This work was supported in part by EPSRC under Grant EP/J004561/1 (BABEL), in part by EPSRC (the U.K. Engineering and Physical Sciences Research Council) under Grant EP/D07908X/1 and Grant EP/G015740/1, in collaboration with the universities of Southampton, Cambridge, and Sheffield and with industry partners ARM Ltd., Silistix Ltd., and Thales, in part by the EU ICT Flagship Human Brain Project under Grant FP7-604102/H2020 720270, in collaboration with many university and industry partners across the EU and beyond, and in part by the European Research Council through the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC under Grant 320689. *(Corresponding author: Alexander D. Rast.)*

A. D. Rast, S. Davidson, S. Davies, M. Hopkins, A. Rowley, A. B. Stokes, and S. Furber are with the School of Computer Science, University of Manchester, Manchester M13 9PL, U.K. (e-mail: adr1r17@soton.ac.uk; simon.davidson@manchester.ac.uk; sergio.davies@gmail.com; michael.hopkins@manchester.ac.uk; andrew.rowley@manchester.ac.uk; alan.barry.stokes@gmail.com; steve.furber@manchester.ac.uk).

S. V. Adams, T. Wennekers, and A. Cangelosi are with Plymouth University, Plymouth PL4 8AA, U.K. (e-mail: samantha.adams@plymouth.ac.uk; thomas.wennekers@plymouth.ac.uk; A.Cangelosi@plymouth.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2816518

I. Introduction: Why Try Spiking Neurorobotics?

ROBOTS provide an interesting and particularly vivid test bed for spiking neural networks. Yet, if the problem to solve does not involve severe time and power constraints, and output fidelity in a fixed task is paramount, a classical robotic solution or an abstract neural simulation will usually produce a better performing, more informative result. But where the seemingly effortless facility of animals to cope with real-world situations suggests lessons to be learned from how the brain does it, spiking neurorobots can be used as a platform for investigating other models of computation.

In cognitive robotics the robot builds a model of behavior based on its own interactions with the real world rather than relying on *a priori* imperative models [1]. Efforts to engineer cognitive neural systems have achieved impressive performance for some real-world tasks [2], and some formal theory exists [3], [4], but truly dynamic behavior has been more elusive [5]. Perhaps biology is doing something different and better, but these models may not be similar enough to the brain to be able to inform the question.

Meanwhile, simulations of brain activity have thus far been semiempirical and as elusive to interpret as their biological prototype [6]. Brain activity is very noisy, depends on probe recording location, and is not exactly replicable from trial to trial [7]. This has left experimenters with a mass of unstructured data by itself revealing few insights into the underlying mental processes taking place [8].

What is needed is a platform that can in some sense extract the computational characteristics that matter from biological neural networks, and be able to apply them in a concrete context that demonstrates *why* they matter, and how we might use them to engineer systems that work with messy real-world data. In this paper, we demonstrate the integration of a "neuromorphic" chip: SpiNNaker, and a complex humanoid robot, the iCub, and show how such a system can learn to recognize and attend to objects of preference in an unsegmented scene, in real time, without relying on off-line training or imperative direction. We further indicate implications to both neuroscience and neural engineering of a structured approach to learning and architectures that might guide design toward autonomous systems. While our neurorobot cannot yet be considered autonomous, we suggest that by demonstrating real-time learning for a simple real-world task, our system

realizes the basic technology and indicates some of the neural design principles that can progress neurorobotics toward fully autonomous behavior.

II. Background: Progress in Neurorobotics

A. Progress in Spiking Neural Models

Spike-based neurorobots might embody behavioral features that are difficult or impossible using other methods [9]. At least in principle, spiking neural networks seem to be able to solve difficult cognitive problems [10] in possibly nonstationary environments [11]. This could inform both neuroscience and robotics [12], however, contrasting priorities between the two communities have created a lack of clarity over what aspects of neural modeling are most meaningful, limiting progress in neurorobotics.

While it is generally agreed that the human brain is divided into functional areas [13], how (or indeed even *if*) populations of neurons are grouped into functional "microcircuit" units remains one of the outstanding problems. Visual [14] and auditory [15] cortex have been well studied, but this is not general. A few "building blocks" have been postulated, notably the winner take all (WTA) [16] circuit, the convolutional network [17], and the reservoir computer [18], but considerable debate exits over their utility or biological plausibility and there is little formal theory to guide further progress.

A more functional approach to brain modeling is described by Eliasmith [4]: interconnection between populations of neurons implement transfer functions and, as such, multiples of these functions implement brain areas [19]. These models have been able to achieve good performance in building working systems with complex behavior, such as Semantic Pointer Architecture Unified Network (SPAUN) [20]. But it is generally thought that the resultant circuits are probably not particularly similar to the brain.

Achieving useful learning with spiking neural networks has long been challenging, and hampered by a lack of theoretical development. One underlying phenomenological Hebbian mechanism: spike-timing-dependent plasticity (STDP) has been comprehensively studied. STDP has been produced in several "flavors": additive [21], multiplicative [22], and trace based [23], subsequently reviewed and improved by others [24], and further extended to a calcium-concentration-based biophysical model [25]. Reasonable network-level models exist for supervised spiking learning [26], [27], but this may not be useful for robots acting in dynamic environments and is not biologically relevant [28]. Fewer models exist for unsupervised learning. Existing approaches have exploited polychronization [29] by applying temporal structure [30] or WTA outputs [28] to solve what are essentially constraint satisfaction problems. However, the more general cognitive problem of learning behavior in a dynamic environment remains.

B. Hardware Progress

A growing number of problems in robotics seem intractably hard without neural hardware [31]. Seeking a solution to unattractive power/performance tradeoffs using conventional computing techniques, cognitive robotics has moved away from abstract, "box based" architectures implemented in traditional hardware [32] toward neural implementations, often integrating "neuromorphic" chips [33] that use specific circuits and architectures derived from neuroscience [34]. Yet despite optimistic predictions about the potential of hardware neural networks applied to robots [35], most implementations of integrated neuromorphic-robotic systems have been limited to proof of concept [36]. Historically, this may have been due to hardware limitations: early chips were probably too small scale [37] for more than very specialized subfunctions. But, a new large-scale generation can credibly implement entire cognitive systems, either in fixed-model analog chips [38] or programmable architectures that can simulate multiple models [39], [40]. Both styles of design emphasize significantly lower power consumption [41], [42] and improved real-time response [17], and feature a variety of learning implementations [12] (not necessarily on-line or on-chip [40]).

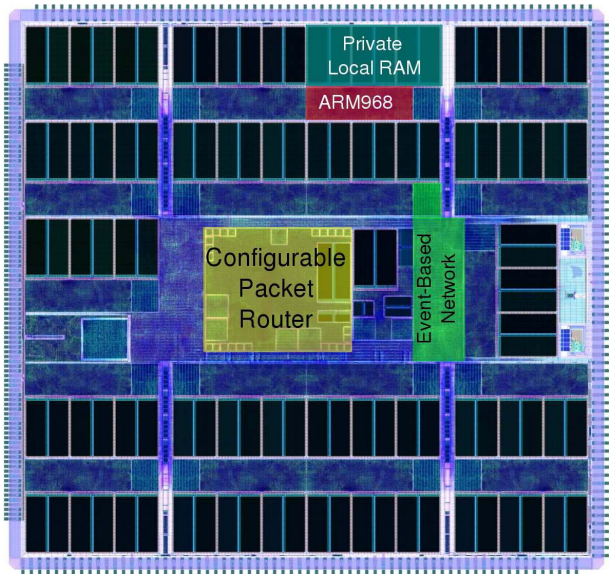


Fig. 1. SpiNNaker chip. Important components are indicated.

While much work has been done on robotic attention, a lot of it concentrates on specific subproblems and few have addressed the problem of on-line learning [43]. Our approach addresses on-line learning directly with embedded hardware. There are advocates both for an "embodied" approach to cognitive robotics where sensing, processing, and actuation, indeed the physics of the body itself, are inseparable [44], and "modular" approaches where a cognitive system functioning more or less in the abstract is bolted onto a robotic body [45]. We feel that to inform learning decisions, identifying, and characterizing modular subsystems that can be simulated prior to physical embodiment, while integrating as much of the sensorimotor periphery directly into the cognitive robotic model (see [46]) as possible, is desirable.

III. Experimental Platform

A. SpiNNaker

The SpiNNaker chip (Fig. 1) is a universal configurable neural network platform designed for real-time simulation.

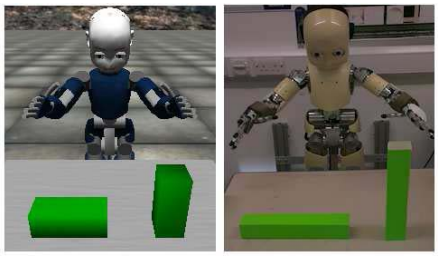


Fig. 2. iCub robot (simulator/actual).

Furber *et al*. [47] and Painkras *et al*. [48] give specific details on the hardware and outline its unique architectural approach. The design goals of SpiNNaker assume *real-time* processing on *real-world* data. Internal timesteps are purely local to a given processor, there is no global memory or support for coherency, and thus there is no "global state" in the sense of an instantaneous system snapshot. To generate an on-chip network from an abstract specification, we use the automated "PACMAN" tool chain that can estimate resource requirements and configure both the neural network and additional support processes such as hardware I/O, application monitoring, and visualization based upon a high-level description in an extended form of the PyNN [49] modeling language.

B. EIEIO

Like many neuromorphic devices, communications on and with SpiNNaker are event based, using a form of address event representation (AER) [50]. Externally, SpiNNaker uses the External/Internal Event Input Output (EIEIO) protocol [51]. EIEIO defines a standardized way of communicating AER data, as well as device-specific or general commands, between possibly heterogeneous platforms. It is a transport-independent protocol layer permitting stateless transceivers which can support any subset of the compete protocol. The current implementation uses user datagram protocol (UDP) packets to bundle spikes into a single packet and send them in "fire-and-forget" manner to a receiving device. Devices identify themselves by UDP port number (possibly shared).

C. iCub

iCub (Fig. 2) is a cognitive developmental robotics platform [52] (http://www.icub.org/) based on a 53 degree-of-freedom humanoid with three-modality (vision/audition/tactile) integrated sensory periphery. Our learning experiments used the iCub platform's included physical simulator, while the other experiments were tested on the real iCub. We use the iCub's standard Yet Another Robot Platform (YARP) protocol (http://wiki.icub.org/yarpdoc/index.html), to transfer information between the iCub and its host PC. Communications between the iCub and SpiNNaker are converted between YARP and EIEIO using a host-based module which acts as a virtual EIEIO device. It maps iCub camera input to the two input layers by generating spikes for each "ON" (white) pixel and receives spikes back from the output layer, representing the fixation location, translating the coordinates of the maximally active neuron in the output layer into iCub view x- and y-coordinates (Fig. 3).

We have also made use of two public external iCub libraries: OpenCV for some basic image processing functions and Aquila–an easy-to-use, high-performance, modular and scalable software architecture for cognitive robotics [53]. In particular, we have used the Aquila modules tracker for extraction of objects from the scene [Fig. 4(c)] and iCubMotor to enable iCub to look and point at salient objects (Fig. 4).

IV. Scalable Model for Object-Specific Attention

A. Biologically Inspired Visual Attention Model

The basis for the model used in this paper was originally described in [54]. This network was subsequently reformulated as a spiking neural network and adapted to run on the SpiNNaker platform [55]. Input came from a spike-based Dynamic Vision Sensor (DVS) camera subsampled to a 16×16 grid but output was virtual: SpiNNaker provided an output signal indicating the preferred position on a 5×5 grid. While the model was functional, it could only handle a single object in the scene (at coarse resolution) and lacked the reinforcement learning system of the original.

In [56], the network was adapted to generate visual attention behavior for the iCub robot. This paper solved the problem of multiple objects, initially using frame-based cameras with frame-to-spike preprocessing over an EIEIO progenitor, AEtheRnet [57], later using DVS input using the "neuromorphic iCub" with EIEIO. Output resolution was improved to 20×20 so that the robot could be made to attend to a live scene, although we observed fairly high sensitivity to lighting and background and jittery response to real-time moving objects. Learning remained disabled.

B. Model Motivation

In this paper, we present an enhanced version of the network, with learning enabled. The model is broadly based on biology but does not attempt an exact replication of still only partially understood brain regions. We have generated a system which includes a bottom-up visual pathway from sensory input to action selection, and a top-down pathway from pre-existing goals to action biassing (Fig. 5). The bottom-up pathway has an input layer representing retinal neurons, coming from hardware or simulated DVS retinas which respond to changes in light level (thus readily detecting, e.g., moving edges). Two different polarities in DVS retinas indicate increasing or decreasing light levels, respectively (onset/offset). When we were simulating DVS we used an onset/offset detector to simulate the separate polarities. Three layers: V1, V2, and V4 represent successive regions in the visual cortex. The top-down pathway has two layers: PFC and frontal eye fields (FEF) representing regions in frontal cortex. Both pathways converge on an output layer: LIP representing a region in parietal cortex. All layers are topographically mapped to the input space so that a neuron represents a fixed visual position in the input image. Layers V1, V2, and V4 are split into a selectable number of orientation-specific sublayers (we used four orientations).

From extensive studies [58], the bottom-up pathway is relatively well understood in terms of functionality as well as

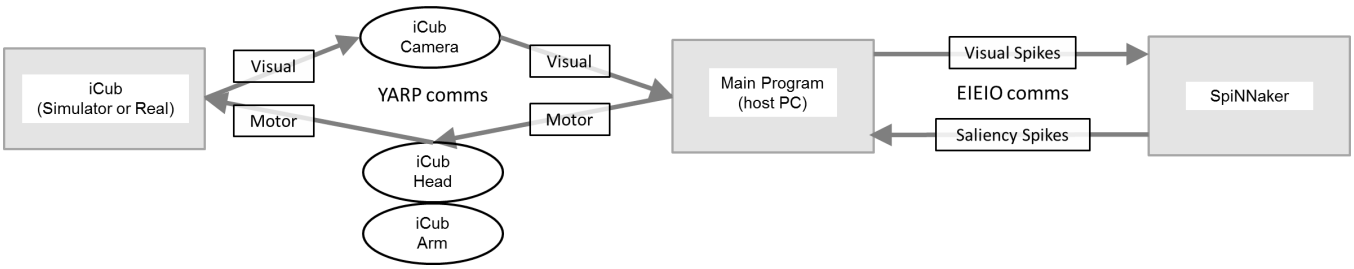


Fig. 3. Combined iCub-SpiNNaker system. Raw I/O from the robot is converted into YARP bottles and processed by a host-based EIEIO transceiver. The transceiver converts the messages into spikes and vice versa, transmitting and receiving them directly to/from SpiNNaker.

connectivity. V1 neurons are considered to be topographically mapped feature detectors, with tuned receptive fields that capture certain basic input features. V2 neurons group local features into larger features such as corners and lines by simple merging of smaller subfeatures from V1, and V4 neurons assemble features into "shapes": complete objects (usually with a closed contour) that can be segmented from a scene (or retinal field). Thus, the feedforward pathway may be seen as a hierarchical object assembly mechanism which identifies (or segments) a scene into objects based upon the features present.

Top-down mechanisms are less completely understood, but in general it is thought that PFC is a center of motivation that directs visual (and other sensory) systems toward goals determined *a priori*, possibly modified by events as they occur. FEF is uniquely involved with visual cortex and is thought to compute a local saliency map and use it to drive attentional output by biassing V4 toward more salient (locally anomalous) objects. The FEF receives projections from V2, and, it is thought, determines saliency using a running spatiotemporal average of local activations. PFC further biases FEF toward goal-relevant stimuli.

LIP is thought to control attention and action selection by targeting basal ganglion (BG) neurons that selectively deinhibit competing action strategies [59]. We have not modeled the BG/striatal system directly (because low-level motor planning is out of scope for the project) but rather use output from LIP to drive the robot's gaze fixation directly toward the active LIP location. One further simplification is that while it is known that FEF directly stimulates LIP as well as V4 (see [60]) we did not implement the FEF-LIP pathway to limit the influence of top-down bias on target selection.

C. Basic Network Model

Neurons in all layers consist of leaky integrate-and-fire (LIF) neurons with current-based exponentially decaying synapses (1). Where synapses are plastic, we use the two-branch exponential STDP additive spike-pair rule modeled after Bi and Poo [21] (2)

$$\left.\begin{aligned} \tau_m \frac{dV}{dt} &= V_r - V + \frac{\tau_m (I_{\text{syn}} + I_{\text{os}})}{C_m} \\ \text{If } V \geq V_t, V &= V_s \tau_{\text{syn}} \frac{dI_{\text{syn}}}{dt} = -I_{\text{syn}} \end{aligned}\right\} \quad \text{LIF equations} \tag{1}$$

Symbol	Meaning	Value
V	Membrane Voltage	(variable)
V_r	Rest Voltage	-65mV
V_s	Reset Voltage	-65mV
V_t	Threshold Voltage	-45mV
I_{syn}	Total Synaptic Current	(variable)
I_{os}	Bias Current (fixed)	0pA
C_m	Membrane Capacitance	1nF
τ_m	Membrane Time Constant	24ms
τ_{syn}	Synapse Time Constant	3ms

$$\delta w = \begin{cases} A_+ e^{-\frac{t_{\text{ps}} - t_{\text{pr}}}{\tau_+}} & \text{if } t_{\text{ps}} > t_{\text{pr}} \\ -A_- e^{-\frac{t_{\text{pr}} - t_{\text{ps}}}{\tau_-}} & \text{if } t_{\text{ps}} < t_{\text{pr}} \\ 0 & \text{if } t_{\text{ps}} = t_{\text{pr}} \text{ or } |t_{\text{ps}} - t_{\text{pr}}| > t_w \\ 0 & \text{if } w = w_{\text{max}} \text{ and } t_{\text{ps}} > t_{\text{pr}} \\ 0 & \text{if } w = w_{\text{min}} \text{ and } t_{\text{ps}} < t_{\text{pr}}. \end{cases} \tag{2}$$

Symbol	Meaning	Value
δw	Weight Change	(variable)
A_+	Potentiating Increment	0.01pA
A_-	Depressing Increment	0.012pA
t_{pr}	Presynaptic Spike Time	(variable)
t_{ps}	Postsynaptic Spike Time	(variable)
τ_+	Potentiating Time Factor	30ms
τ_-	Depressing Time Factor	30ms
t_w	STDP Time Window	24ms
w_{max}	Maximum Weight	20 pA
w_{min}	Minimum Weight	0 pA

A time window is used to avoid having to keep an unbounded spike record in memory and stipulates that beyond a certain difference in time between spike pairs, the contribution to the weight change is negligible. Maximum and minimum weights prevent unbounded weight change under the additive rule, and the minimum also prevents synapses (biologically unrealistically) "flipping" between excitatory and inhibitory.

Layers produce a progressive user-selectable subsampling of the input. In our experiments V1 and V2 layers were notionally subsampled at $1/1.6$ input width in either X- or Y-dimensions, PFC, V4, and LIP at $1/2$ the V2 width. As noted, each major layer except LIP is subdivided into four orientated sublayers, representing orientations $\{0, (\pi/4), (\pi/2), (3\pi/4)\}$, respectively.

Each V1 sublayer receives input from a neighborhood of input neurons around its topographic position using a series of tuned orientation filters that provide a receptive field. These

Fig. 4. Stages of preprocessing in the iCub input chain for presentation to the network used in tests. (a) Raw image is transformed first to a (b) "saturation" view, then tracked to a (c) minimal bounding region. This bounding region is (d) subsampled to a final input matrix, each location representing a coordinate neuron in the inputs to V1.

filters use a module that can generate Gaussian, Gabor, or normal/inverted Mexican-hat receptive fields in a variety of scales, eccentricities, and orientations. Weights implement the receptive field strength based on the distance of the neighboring input neuron from its V1 target. Thus neurons near the target will have high connection weights (or low ones in the case of the inverted Mexican-hat filter) and more distant ones will have correspondingly lower weight, further biased by orientation. In our experiments, we used 2-D Gaussian receptive fields with an eccentricity (ratio of major to minor axis) of 5.5 in two different scales with a base tile size of 5, to simulate orientated line detectors. Each scale is a multiple of the tile size so that the filters are in 5×5 and 10×10 input neurons, overlapping across the visual field. Each tile projects to a single neuron in the V1 layer at the corresponding position, taking the 32×32 input and subsampling at a ratio of $1/1.6$ to get 20×20 of each scale of tile.

V2 neurons receive input from a neighborhood of V1 neurons using a simple pooling model that establishes a region, with tunable size and weight value, around the V2 target from which V1 neurons will project with identical weights. In our experiments, since V2 had the same number of neurons as V1 we set the region to be a single neuron, i.e., a V2 neuron receives input from one 5×5 V1 tile and one 10×10 tile. Each V2 sublayer has internal lateral inhibition in a tunable local radius around each neuron providing a form of soft WTA competition. We set this radius to 2, i.e., the size of the WTA pooling in V2 is 5×5. An optional global WTA filter in the V2 layer, which we enabled in our experiments, permits additional competition between sublayers (i.e., between orientations as well as within pools in a given orientation).

V4 neurons receive input from each of the V2 neurons in a locally subsampled region of V2. Thus if the subsampling is $1/2$, the V4 neuron will process a subsampled 2×2 patch of the V2 space. In the initial version of the network, with hardwired PFC bias, each V4 neuron in each sublayer received bias input from all PFC neurons in the same sublayer. In later tests, as will be seen, the PFC includes additional mapping, each PFC neuron biasing its corresponding V4 neuron(s) by both position and orientation (with possible subsampling/oversampling).

LIP neurons merge input from V4 neurons for each orientation in a one-to-one correspondence, where each sublayer in V4 maps by position to the corresponding LIP neuron. The LIP also includes an internal hard WTA to select a single attentional position at each moment, each neuron inhibiting all neurons in the population, including itself, but with higher weight for nonself-connections than for the self-connection.

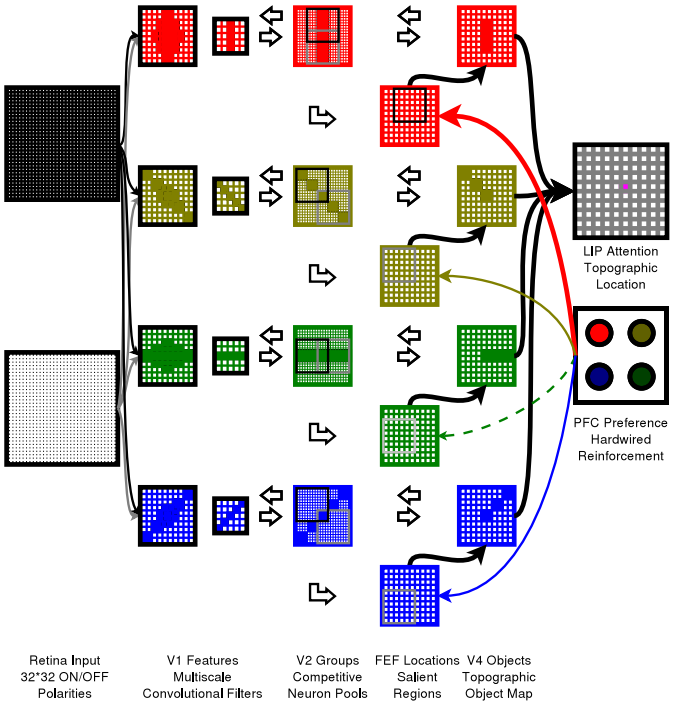


Fig. 5. Enhanced network. This shows the network as it is with all enhancements turned ON. The input retina layer is a real or simulated visual field taken either from the preprocessed robot imaging system or from a software image generator. Each of layers V1, V2, V4, and prefrontal cortex (PFC) are separated into four orientations per layer. Layer lateral intraparietal cortex (LIP) merges orientations via a WTA. Except for the input retinal layer, the PFC layer and the output LIP layer, the diagram shows one "tile" representing a particular topographic location from the retinal field; tiles extend over the entire visual field. Each of the large and small boxes in V1 represents a different scale of convolutional filter. In V2, the internal black and gray boxes represent a mapping from one of the smaller V1 filters. The black one represents the filter shown in the diagram, while gray represents another filter (not shown). Open "wide" arrows represent connections that are understood to extend over all tiles in a layer but to connect pairs of tiles at the same topographic position in each layer. Closed "narrow" arrows represent one-to-one connections between specific neurons in their associated layers with strength given by linewidth. Feedback connections project in each case so that each actual synaptic link established is bidirectional.

How this network is intended to work in principle is like this: the V1–V2–V4 pathway selects progressively sharpened locations of visual interest. If the V2 WTA is enabled then the network is encouraged to select a single most-salient location. PFC then biases the V4 layer to prefer objects lying in one orientation and respond aversively toward objects in another orientation. The biasing effect should produce a stronger input to the LIP neurons in the preferred location of visual interest in V4 and the LIP WTA should then select the appropriately preferred location. Overall behavior is a "goal-directed object selector": a system that recognizes objects in an initially undifferentiated input scene and directs attention toward objects of interest.

D. Enhanced Visual Attention Model

In an effort to solidify behavioral reliability, before moving to learning experiments, we made a series of architectural enhancements to the network. Each enhancement is an optional parameter that can be added incrementally to observe the isolated effect of each as well as the cumulative effect of them. For complete detail, we refer readers to the original PyNN source in the Appendix and here describe the features of interest.

Upscaled Resolution and Size The original model subsampled the native DVS 128×128 resolution to 32×32. With larger SpiNNaker systems and enhanced interfaces available, we added a scaling module which permits various input sizes. We have tested this at 32×32, 64×64, and 128×128 resolutions (using both frame-based and DVS cameras).

Interlayer Feedback Experiments on convolutional networks similar to the input stages (V1, V2, V4) suggest that a more biologically realistic recurrent topology with feedback between layers should enhance contrast and enable attention to be maintained during periods of object overlap. We therefore added feedback with tunable strength, typically set to 0.8 of the feedforward weight (later marginally tuned to 0.81), between V4 and V2, and between V2 and V1. Feedback paths project in an inverse pattern of the forward projections. With a subsample ratio of $1/2$ each V4 neuron thus projects to four V2 neurons, while since the ratio of V1-V2 neurons is 1:1, each V2 neuron projects back to a single V1 neuron.

FEF-Like Layer To improve bias specificity so as to target neurons in the visual field where stimulus is present, we replaced the hardwired PFC with a more biologically realistic topographically mapped FEF layer that computes a top-down attentional bias based on the expected input. Following [60], we connected V2 to FEF using orientated 2-D Gaussian filters with a subsample ratio of $1/2$, similar to the filter connections used between the input and V1. The FEF projects to V4 using one-to-one connectors which map each FEF output to its corresponding neuron in each V4 orientation. This input represents a dynamic prediction of the subregions in the visual field where V4 should expect "interesting" (strongly orientated) input, based on recent activity. The PFC projects to FEF in topographically mapped one-to-one connections using N-methyl-D-aspartate (NMDA)-like synapses with long time constants to provide a source of quasi-persistent bias. These connections selectively gate the input from the bias source so that bias can be "switched out" when the network has learned the orientation preference. To enhance bias contrast, we replaced the excitatory-only PFC output with bipolar output using a relay layer of inhibitory neurons to convert FEF output in the aversive orientation to inhibitory V4 input.

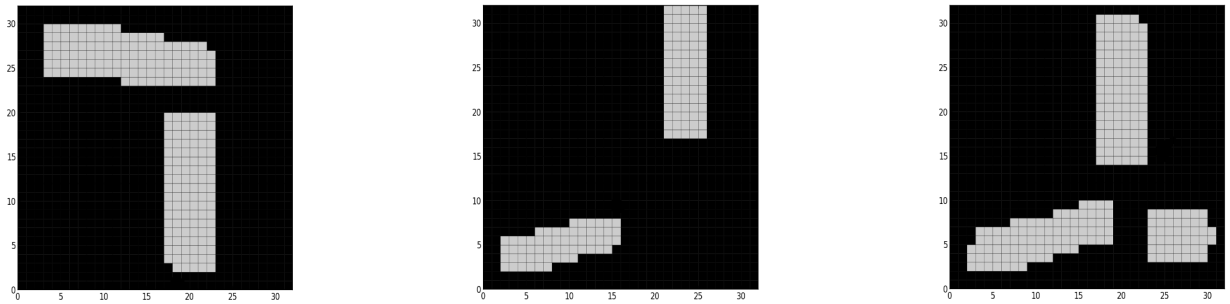
Learning By design, the network has been constructed so that the relative timing between firings in V2 and V4, and hence the weights between these layers, has the greatest impact on task performance. We enabled STDP between V2 and V4 using PyNN's SpikePairRule with AdditiveWeightDependence (for simplicity of analysis) in a time window of ± 30 ms. The additive terms are slightly asymmetric, potentiative increment being set to 0.01 nA, depressive decrement to -0.012 nA, from a weight range of $[0, 20]$ nA. Output from FEF was tuned to provide subthreshold stimulation to V4: sufficient to drive V4 neurons in the preferred orientation to a regime just below spiking, but not to cause spontaneous spikes. Likewise the FEF inhibits V4 neurons in the aversive orientation sufficiently to prevent spiking on a single spike from V2, but not so much as to suppress spiking altogether. Thus, we expect V2 neuron firings to be strongly causal with respect to V4 firings in the preferred orientation and only weakly causal in the aversive orientation. In such a system STDP should enhance contrast and bias the network toward the preferred orientation.

V. Results

A. Feedback in a Convolutional Network

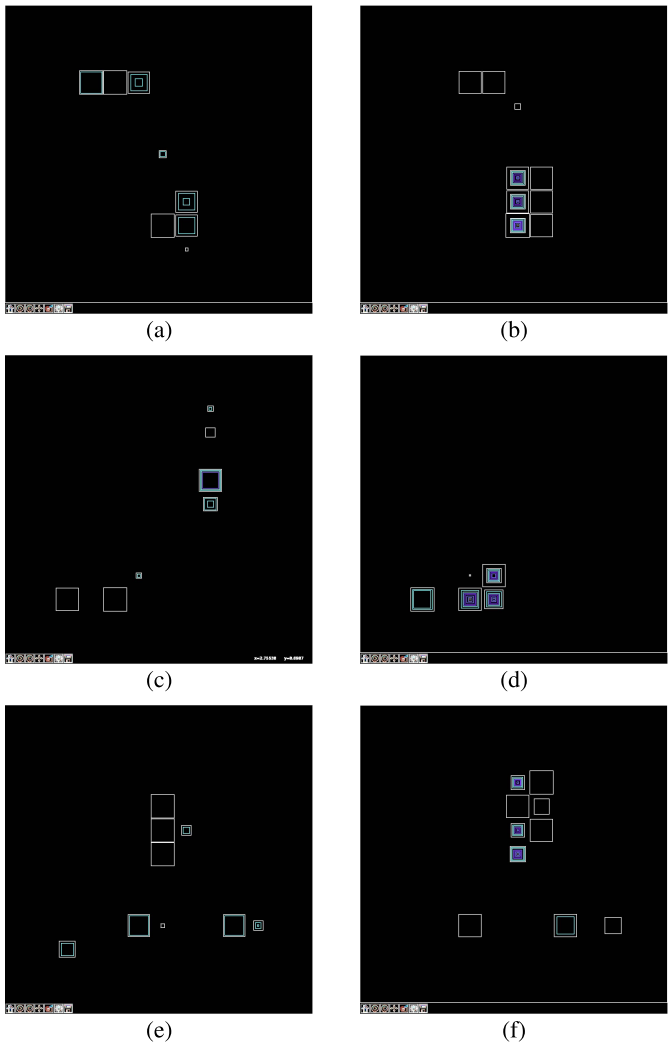
In Section IV-D, we mentioned the enhancement of the original network by enabling feedback between layers. To test the effect of feedback, we ran simulations using three different stimuli (Fig. 6) with feedback enabled and with feedback disabled, at 32×32 input resolution with no other enhancements. With feedback disabled the optimal weight value from $V2 \rightarrow V4$ was 5.5 nA, whereas with feedback on the optimal value was 4.5 nA, as expected since feedback causes slightly higher activation in each of its affected layers.

Fig. 7 shows the difference in behavior with feedback on and feedback off. Each box represents a spike in the output neuron at the corresponding position in the LIP output map. Boxes nest in increasing time with a resolution of 1 ms and a maximum spike frequency of 333.3 Hz. Larger boxes represent earlier times in the simulation. The largest possible box would be the size of an entire LIP output tile and indicate a spike at 0 ms, the smallest a dot and represent a spike at the end of simulation time, e.g., 100 ms in a 100 ms simulation. A leaky integrator colors the boxes so that white represents low levels of recent prior spiking and purple high levels. Thus, these figures [Figs. 7(a), (c), and (e), 8(a), (c), and (e), 9(a), (c), and (e), 11(a) and (c), and 12(a) and (c)] represent a mapped record of actual stimulus to eye-motion activators. For each of three different inputs, enabling feedback enhances contrast and greatly increases the activation in the preferred direction (sometimes with a slight increase in the aversive direction). In the case of Stimulus 2, indeed, it eliminated a consistent pattern-sensitive error in the case of horizontal preference.



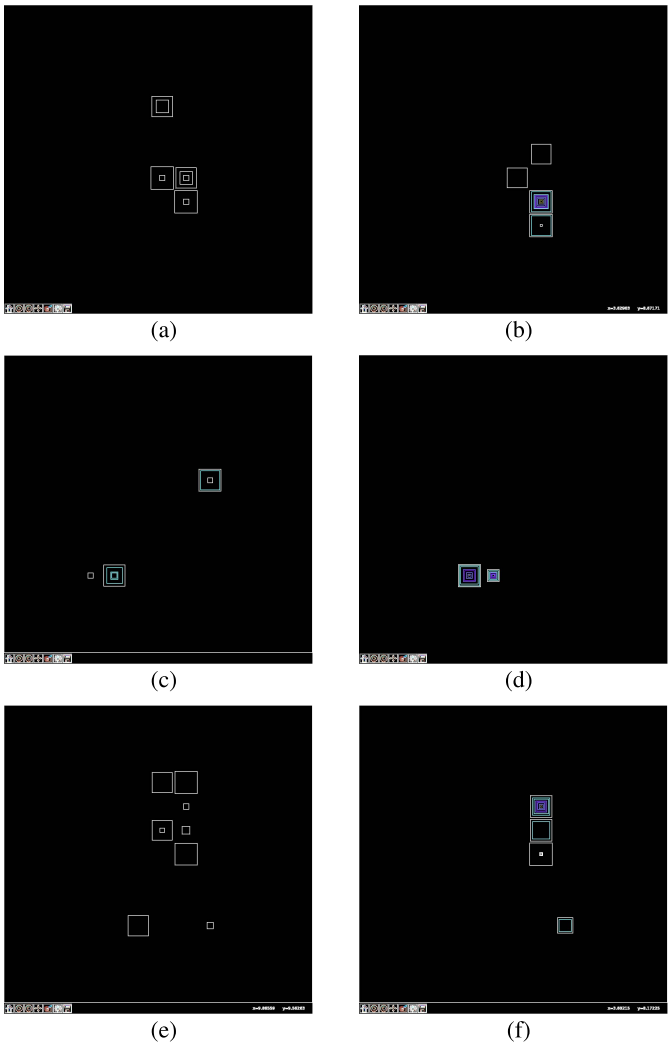
(a) (b) (c)

Fig. 6. Stimuli used for testing the effects of various network enhancements. These views show the subsampled inputs as seen after preprocessing following the chain shown in Fig. 4. Each stimulus has an horizontal and a vertical object. The third stimulus also has a "distractor" ball object with no clear orientation. The positions of these objects correspond to the same positions in the topographic output spike plots in Figs. 7 and 8. (a) Stimulus 1. (b) Stimulus 2. (c) Stimulus 3.



(a) (b) (c) (d) (e) (f)

Fig. 7. Comparison between network activations with feedback disabled and enabled, respectively. (a) Stimulus 1, feedback off, vertical preferred. (b) Stimulus 1, feedback on, vertical preferred. (c) Stimulus 2, feedback off, horizontal preferred. (d) Stimulus 2, feedback on, horizontal preferred. (e) Stimulus 3, feedback off, vertical preferred. (f) Stimulus 3, feedback on, vertical preferred.



(a) (b) (c) (d) (e) (f)

Fig. 8. Network performance with targeted bias using the FEF-like layer alone and in combination with feedback. (a) Stimulus 1, FEF on, vertical preferred. (b) Stimulus 1, FEF and feedback on, vertical preferred. (c) Stimulus 2, FEF on, horizontal preferred. (d) Stimulus 2, FEF and feedback on, horizontal preferred. (e) Stimulus 3, FEF on, vertical preferred. (f) Stimulus 3, FEF and feedback on, vertical preferred.

B. Targeted Bias Using a FEF-Like Layer

We next investigated the effect of creating targeted bias from a FEF-like layer rather than overall global bias per orientation. With feedback disabled (Fig. 8), as predicted in Section IV-D, the network now has a sharpened pattern of fixations, with fewer locations going active. Notably, like the feedback enhancement, with this one feature (FEF) enabled,

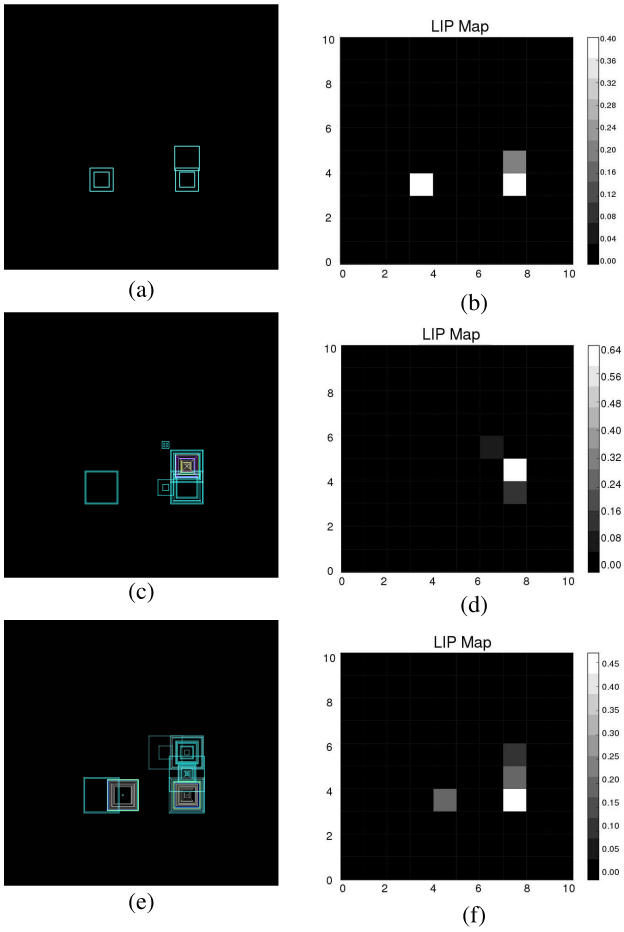


Fig. 9. Results for learning on the iCub with a two-stimulus scene. PFC is enabled, vertical objects set to preferred. The box-plot diagrams have the same interpretation as shown in Fig. 7. The salient location shows each position's integrated spike count. (a) 50-ms run, mapped spike output. (b) 50-ms run, salient location. (c) 200-ms run, mapped spike output. (d) 200-ms run, salient location. (e) 1000-ms run, mapped spike output. (f) 1000-ms run, salient location.

the network did not exhibit erroneous fixations on the aversive object or away from objects.

Although the fixation patterns obtained using the FEF were encouraging, they are still fairly sparse and would cause relatively feeble drive to robot actuators, thus small, slow movements. But with feedback as well as the FEF enabled, the results (Fig. 8) were dramatic. Fixation rate and robustness of spiking is improved significantly; typically one neuron directly on the preferred object fires in overwhelming preference to other locations, resulting in stable and rapid fixation. This would allow the robot to be capable of saccade-like shifts of attention to the target object, and even the presence of distractors (see the results for Stimulus 3) does not significantly affect fixation performance. With this set of enhancements the robot has been taken from an ability to fixate on a general region with some attentional wandering to immediate focus on a target object.

C. Learning Results

We ran a series of trials with the simulated iCub using two stimuli, one horizontal and one vertical, as seen from Fig. 4.

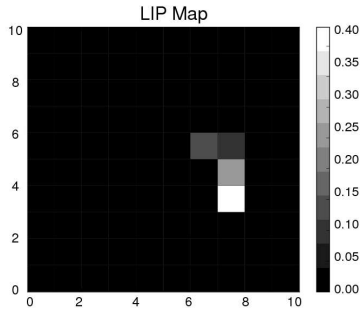


Fig. 10. Salient location for the postlearning trial [for the prelearning trial refer to Fig. 9(b)]. PFC and learning disabled.

We set PFC bias on throughout the trials and ran this network progressively for 50, 100, 200, 500, and 1000 ms, respectively (Fig. 9). Fixation performance increases throughout the trials so that for longer run lengths the robot strongly prefers the "preferred" stimulus (vertical for the figures shown). We then ran the same tests for a 50 ms run with PFC biasing disabled (and with plasticity off) and observed as seen from Fig. 10 that the network attends immediately to the preferred stimulus, in contrast to the 50-ms case before learning, where even with PFC bias on, fixation performance is poor.

We next considered the impact of a distractor object. We used a scene [as seen schematically in Fig. 6(c)] with both target objects and a "neutral" distractor: a round ball with no definite orientation. We then ran learning trials for both orientations for 1000 ms, and tested the recall of the learned orientation with the distractor present. Figs. 11 and 12 show the results. As can be seen, the correct target object remains the focus of attention despite the presence of distractors.

VI. Discussion

An examination of the weight results after learning is instructive. We generated topographic plots of the weight changes after learning for each of the trials for both positive (potentiating) and negative (depressing) changes (Fig. 13). As expected, positive changes dominate and affect only the preferred orientation (with no weight changes in the aversive direction). By design, the topographic pattern of V2 projections to V4 makes a V2 spike much more likely to be causal than anticausal with respect to its associated V4 neuron firings, making potentiation more likely than depression. Meanwhile, bipolar biasing from the PFC/FEF loop activates the V4 spikes necessary to trigger STDP in the preferred direction while suppressing those in the aversive direction. However, connections related to *all* the objects in the scene get strengthened, not just those associated with the preferred stimulus. One possible explanation is that because objects have finite size in both dimensions, they trigger some level of activation in all other orientations. Finally, we observed that the horizontal preference generated stronger overall patterns of weight modification than the vertical direction. This may be due to the distortion by perspective of 3-D objects projected onto the 2-D retinal field, which causes the vertical edges to be somewhat diagonal and hence triggering mixed activations.

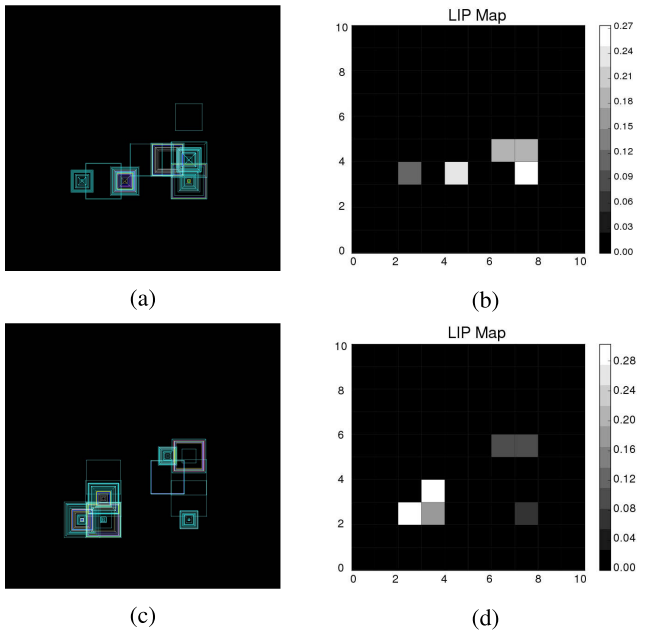


Fig. 11. Results for learning on the iCub with a two-stimulus scene in the presence of a distractor. STDP was enabled and the PFC provides a source of bias via LIP as indicated by the directional preference. The network is configured as for the tests in 9 and output results are in the same pair of formats. (a) Learning trial, vertical object preferred, output spikes. (b) Learning trial, vertical object preferred, salient location. (c) Learning trial, horizontal object preferred, output spikes. (d) Learning trial, horizontal object preferred, salient location.

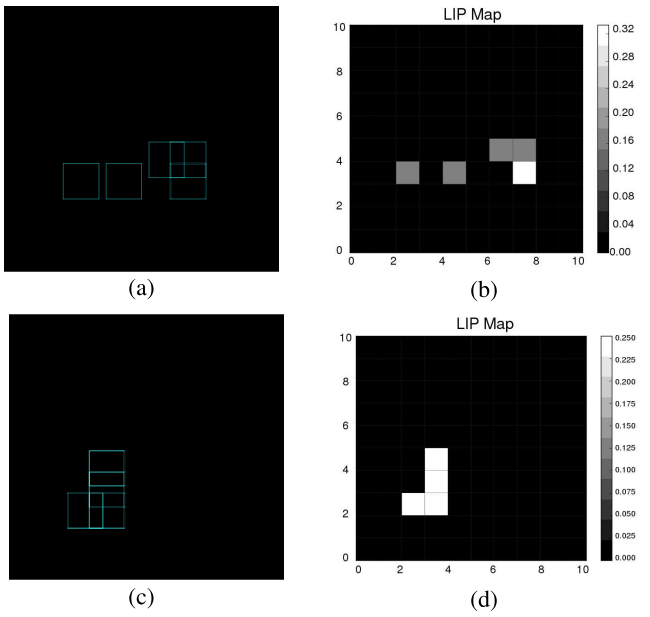


Fig. 12. Results for recall on the iCub with a two-stimulus scene in the presence of a distractor. STDP was off and the PFC did not provide any top-down attentional bias. Otherwise, the network is configured as for the tests in 11. (a) Recall trial, vertical object preferred, output spikes. (b) Recall trial, vertical object preferred, salient location. (c) Recall trial, horizontal object preferred, output spikes. (d) Recall trial, horizontal object preferred, salient location.

All these effects come from the same common root cause: the need for pre-post spike pairings in order to trigger any sort of weight update. One might expect that a learning rule triggered on either pre- or postsynaptic spike without the need for pairing (as in classical STDP) might enhance results further and produce still better contrast–and indeed preliminary modeling experiments (not reported here) have shown results that are encouraging.

When we turn to network structure and look at the effect of successive enhancements, we find a striking additive characteristic. Each change, taken in isolation, tends to result in modest improvements in performance, in some cases transforming a nonfunctional network into one that could successfully perform the task. But the impact on behavior could be subtle and open to question. By contrast when *all* the enhancements were switched on the behavior improved to the point that it is convincing (Fig. 14). This is consonant with biology, suggesting a series of point mutations produces a modular network robust against component failure rather than an undifferentiated pool of neurons for the most part lacking meaningful structure. While it is at this point too early to make any definite conclusions, it is possible that one reason neurorobotics has had difficulty in matching more conventional imperative methods is simply that previous networks have had architectures that try to isolate the impact of a particular network feature.

The results that we have obtained demonstrate that a systematic program of enhancements on a well-characterized network can result in dramatic improvements in behavioral performance. We were able to achieve this with an approach that leverages the nature of STDP: a causality-based learning

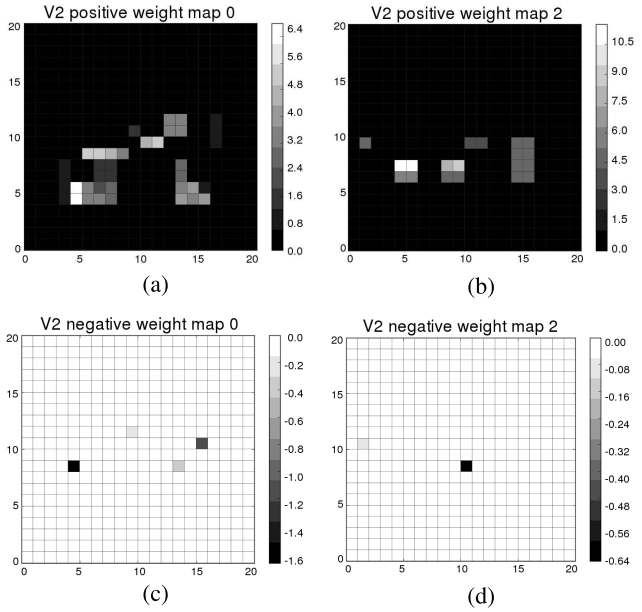


Fig. 13. Weight changes for the learning experiments conducted in 11, for the sublayer indicated by the orientation preference. There were no weight changes in the other (nonpreferred) sublayers in either case. (a) Positive weight changes, horizontal preference. (b) Positive weight changes, vertical preference. (c) Negative weight changes, horizontal preference. (d) Negative weight changes, vertical preference.

rule. By ensuring that PFC/FEF input into V4 makes it more probable that the correct V4 neurons will spike upon input from V2, we can reduce the sensitive pathway to the $V2 \rightarrow V4$ connection and hence reliably instantiate learning on that layer with expected improvement in performance.

Unexceptional performance can lead to an overly gloomy assessment of the potential of neural networks to solve

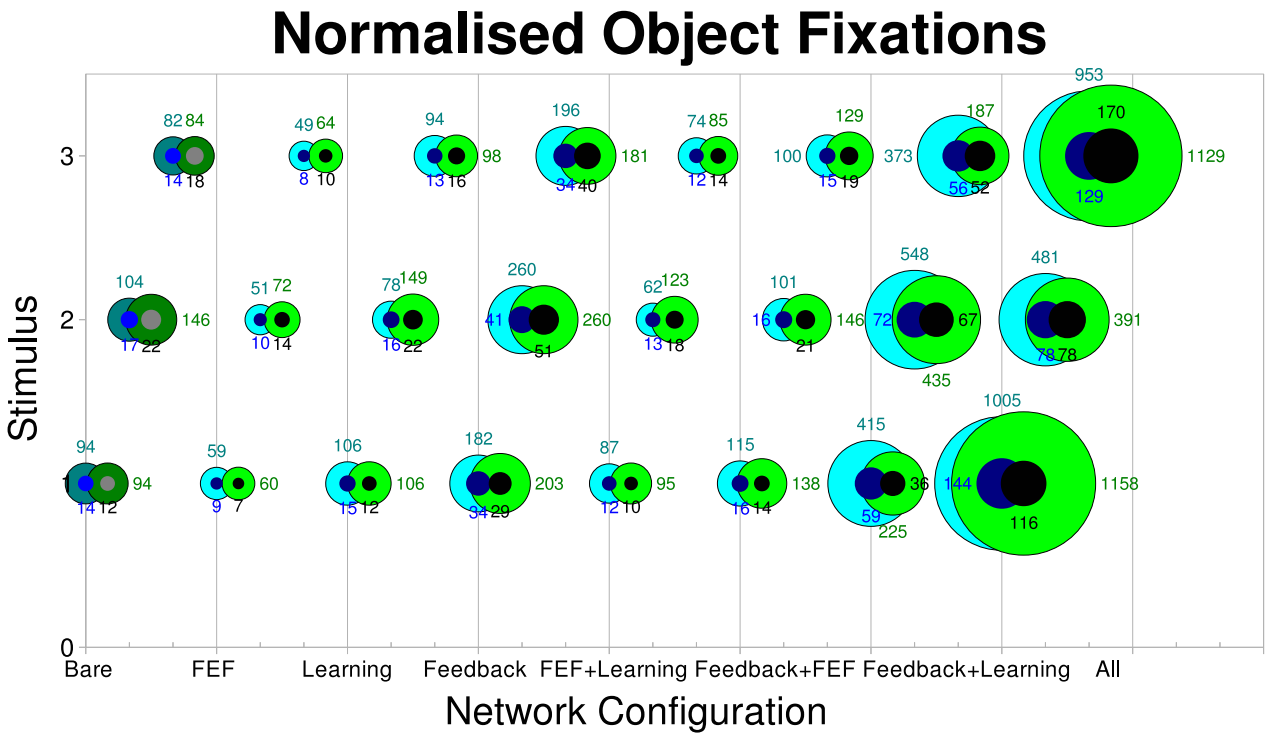


Fig. 14. Effects of adding various enhancements to the model. Each data bubble reflects the object fixations for 10 separate trials with the same stimulus and the same direction chosen as preferred. Cyan bubbles represent horizontal preferred, green vertical preferred. The dark bubbles represent the object fixations on the aversive object. Each row represents one stimulus, each columnar region one network configuration as shown. Numbers beside the bubbles, color matched to their associated point, represent the absolute fixation counts, normalized to the size of the object.

real-world problems [61], [62], but is itself complicated by the difficulty in establishing good metrics. Most approaches to date rely to some extent on ad hoc techniques. Indeed, the methods we used in deriving Fig. 14, although following techniques well-established from eye fixation studies [63] require an ad hoc determination of whether a given fixation point is within the target object or outside it. We experimented with an approach based on mutual information but this breaks down due to the presence of too many zeroes in the input distributions. DeGroot [64] has suggested a method based on increase in utility but this still leaves the question of deciding the utility function itself. We would like to use the experimental results reported here as a stimulus for more mathematically rigorous formal theoretical models able to characterize quantitatively the properties needed to build biologically realistic networks whose performance can be specified directly.

VII. Conclusion

In the final analysis, what is being researched? Is it the neuroscience of the brain, or is it the engineering of functional robots? The cognitive neurorobotics approach allows both to be pursued in the same context. As we have done here it can be used as a tool to uncover the model of computation, and then in a recursive process take the insights thus gained to refine the model systematically and produce systems that function in the real world. We find several features. First, it appears that contrast enhancement is one of the most important functions of recurrent feedback. The effects of recurrent connections in the signal-processing V1–V2–V4 pathway, in the PFC–V2–V4 pathway, and indeed in WTA structures in both V2 and LIP, all acted to enhance contrast. Second, in contrast to the largely "selective," fixed-function nature of conventional digital processing, neural structures appear to operate in a "cumulative" fashion where additional modules or connections can be recruited to enhance performance of a specific function, without any one being critical. Third, neural networks appear to function best when the quiescent input for most neurons in a layer puts them just below the spiking threshold. It was notable that PFC biasing in V2 and V4 "primes" the neurons for firing when a spike arrives and makes it more probable that learning will be triggered. Finally, classical STDP is an important mechanism for learning positive causation but is almost certainly less important in learning anticausation. Other learning mechanisms are probably responsible for anticausal learning and this may explain why learning experiments using STDP alone have been notoriously challenging to make functional. However, in a larger sense this might summarize our overall conclusion: **spiking neural networks that rely on a single structure or effect to produce results will probably perform unspectacularly; spiking networks that employ a combination of effects can probably be made to perform convincingly.** It remains for the future to be seen how this combination of effects can be synthesized into an overall model of computation for the brain.

Appendix
PyNN Code for the Networks Used in This Paper

The complete PyNN scripts for the networks used are available at the following: https://github.com/SpiNNaker Manchester/BehaviouralLearning/. A readme file describes

how to use the files. These scripts may be run using SpiNNaker from the human brain project portal site at: http://collaboration.humanbrainproject.eu.

The source data for the simulations run in the tests is also available on the same site. These are grouped into /enhancements, /scaling, and /learning folders for easy reference. Using the utility package spike_file_to_spike_array users can run sample inputs as generated by the iCub. Each of the source data folders includes the original graphics for the box plots shown in this paper, for ease of on-screen readability.

ACKNOWLEDGMENT

SpiNNaker has been 15 years in conception and 10 years in construction. The authors would like to thank many and varied folk in Manchester and in various collaborating groups around the world for their contributions to get the project to its current state.

REFERENCES

[1] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robot. Auto. Syst.*, vol. 37, nos. 2–3, pp. 185–193, Nov. 2001.

[2] Y. Sandamirskaya and G. Schöner, "An embodied account of serial order: How instabilities drive sequence generation," *Neural Netw.*, vol. 23, no. 10, pp. 1164–1179, Dec. 2010.

[3] G. Schöner, M. Dose, and C. Engels, "Dynamics of behavior: Theory and applications for autonomous robot architectures," *Robot. Auto. Syst.*, vol. 16, nos. 2–4, pp. 213–245, Dec. 2005.

[4] C. Eliasmith, "A unified approach to building and controlling spiking attractor networks," *Neural Comput.*, vol. 17, no. 6, pp. 1276–1314, Jun. 2005.

[5] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen. (Apr. 2016). "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection." [Online]. Available: https://arxiv.org/abs/1603.02199

[6] T. C. Potjans and M. Diesmann, "The cell-type specific cortical microcircuit: Relating structure and activity in a full-scale spiking network model," *Cereb. Cortex*, vol. 24, no. 3, pp. 785–806, Mar. 2014.

[7] F. Wood, M. J. Black, C. Vargas-Irwin, M. Fellows, and J. P. Donoghue, "On the variability of manual spike sorting," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 912–918, Jun. 2004.

[8] S. Todorova, P. Sadtler, A. Batista, S. Chase, and V. Ventura, "To sort or not to sort: The impact of spike-sorting on neural decoding performance," *J. Neural Eng.*, vol. 11, no. 5, p. 056005, Oct. 2014.

[9] Y. Kuniyoshi and L. Berthouzea, "Neural learning of embodied interaction dynamics," *Neural Netw.*, vol. 11, nos. 7–8, pp. 1259–1276, Oct./Nov. 1998.

[10] N. Kasabov and E. Capecci, "Spiking neural network methodology for modelling, classification and understanding of EEG spatio-temporal data measuring cognitive processes," *Inf. Sci.*, vol. 294, pp. 565–575, Feb. 2015.

[11] E. Smith and M. Lewicki, "Efficient coding of time-relative structure using spikes," *Neural Comput.*, vol. 17, no. 1, pp. 19–45, Jan. 2005.

[12] F. Walter, F. Röhrbein, and A. Knoll, "Neuromorphic implementations of neurobiological learning algorithms for spiking neural networks," *Neural Netw.*, vol. 72, pp. 152–167, Dec. 2015.

[13] Y. He, Z. J. Chen, and A. C. Evans, "Small-world anatomical networks in the human brain revealed by cortical thickness from MRI," *Cerebral Cortex*, vol. 17, no. 10, pp. 2407–2419, Oct. 2007.

[14] T. Poggio and T. Serre, "Models of visual cortex," *Scholarpedia*, vol. 8, no. 4, p. 3516, 2013.

[15] L. M. Aitkin, *Auditory Cortex: Structural and Functional Bases of Auditory Perception*, 1st ed. Dordrecht, The Netherlands: Springer, Jul. 1990.

[16] M. Oster and S.-C. Liu, "A winner-take-all spiking network with spiking inputs," in *Proc. 11th IEEE Int. Conf. Electron. Circuits Syst. (ICECS)*, Dec. 2004, pp. 203–206.

[17] R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jimenez, C. Serrano-Gotarredona, and J. A. Perez-Carras, "On real-time AER 2-D convolutions hardware for neuromorphic spike-based cortical processing," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1196–1219, Jul. 2008.

[18] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.

[19] C. Eliasmith, *How to Build a Brain: A Neural Architecture for Biological Cognition*. London, U.K.: Oxford Univ. Press, Jun. 2015.

[20] T. C. Stewart, F.-X. Choo, and C. Eliasmith, "A perception-cognition-action model using spiking neurons," in *Proc. Cognit. Sci. Soc.*, 2012, pp. 1018–1023.

[21] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, Dec. 1998.

[22] J. Rubin, D. D. Lee, and H. Sompolinsky, "Equilibrium properties of temporally asymmetric Hebbian plasticity," *Phys. Rev. Lett.*, vol. 86, no. 2, pp. 364–367, Jan. 2001.

[23] A. Morrison, A. Aertsen, and M. Diesmann, "Spike-timing-dependent plasticity in balanced random networks," *Neural Comput.*, vol. 19, no. 6, pp. 1437–1467, Jun. 2007.

[24] H. D. Abarbanel, R. Huerta, and M. I. Rabinovich, "Dynamical model of long-term synaptic plasticity," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 15, pp. 10132–10137, Jul. 2002.

[25] M. Graupner and N. Brunel, "Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location," *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 10, pp. 3991–3996, Mar. 2012.

[26] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," *Neural Comput.*, vol. 22, no. 2, pp. 467–510, Feb. 2010.

[27] Y. Xu, X. Zeng, L. Han, and J. Yang, "A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks," *Neural Netw.*, vol. 43, pp. 99–113, Jul. 2013.

[28] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Front Comput. Neurosci.*, vol. 9, pp. 99–113, Aug. 2015.

[29] E. M. Izhikevich, "Polychronization: Computation with spikes," *Neural Comput.*, vol. 18, no. 2, pp. 245–282, Feb. 2006.

[30] B. Rekabdar, M. Nicolescu, R. Kelley, and M. Nicolescu, "An unsupervised approach to learning and early detection of spatio-temporal patterns using spiking neural networks," *J. Intell. Robot. Syst.*, vol. 80, pp. S83–S97, Dec. 2015.

[31] F. Alnajjar, I. B. M. Zin, and K. Murase, "A hierarchical autonomous robot controller for learning and memory: Adaptation in a dynamic environment," *Adapt. Behav.*, vol. 17, no. 3, pp. 179–196, 2009.

[32] A. Kirsch, "Robot learning language—Integrating programming and learning for cognitive systems," *Robot. Auto. Syst.*, vol. 57, no. 9, pp. 943–954, Sep. 2009.

[33] J. Conradt, F. Galluppi, and T. C. Stewart, "Trainable sensorimotor mapping in a neuromorphic robot," *Robot. Auto. Syst.*, vol. 71, pp. 60–68, Sep. 2015.

[34] M. A. C. Maher, S. P. DeWeerth, M. A. Mahowald, and C. A. Mead, "Implementing neural architectures using analog VLSI circuits," *IEEE Trans. Circuits Syst.*, vol. 36, no. 5, pp. 643–652, May 1989.

[35] F. Stramandinoli, D. Marocco, and A. Cangelosi, "The grounding of higher order concepts in action and language: A cognitive robotics model," *Neural Netw.*, vol. 32, pp. 165–173, Aug. 2012.

[36] K. Shimonomura, T. Kushima, and T. Yagi, "Binocular robot vision emulating disparity computation in the primary visual cortex," *Neural Netw.*, vol. 21, nos. 2–3, pp. 331–340, 2008.

[37] A. G. Andreou, K. A. Boahen, P. O. Pouliquen, A. Pavasović, R. E. Jenkins, and K. Strohbehn, "Current-mode subthreshold MOS circuits for analog VLSI neural systems," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 205–213, Mar. 1991.

[38] J. Schemmel, D. Brüderle, A. Grübl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proc. IEEE Int. Symp. Circuit. Syst. (ISCAS)*, May/Jun. 2010, pp. 1947–1950.

[39] A. Rast *et al.*, "Concurrent heterogeneous neural model simulation on real-time neuromimetic hardware," *Neural Netw.*, vol. 24, no. 9, pp. 961–978, Nov. 2011.

[40] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.

[41] P. Livi and G. Indiveri, "A current-mode conductance-based silicon neuron for address-event neuromorphic systems," in *Proc. IEEE Int. Symp. Circuit. Syst. (ISCAS)*, May 2009, pp. 2898–2901.

[42] E. Stromatias, F. Galluppi, C. Patterson, and S. Furber, "Power analysis of large-scale, real-time neural networks on SpiNNaker," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–8.
[43] M. Begum and F. Karray, "Visual attention for robotic cognition: A survey," *IEEE Trans. Auto. Mental Develop.*, vol. 3, no. 1, pp. 92–105, Mar. 2011.
[44] L. A. Stein, "Postmodular systems: Architectural principles for cognitive robotics," *Cybern. Syst.*, vol. 28, no. 6, pp. 471–487, Sep. 1997.
[45] O. Sporns, N. Almássy, and G. M. Edelman, "Plasticity in value systems and its role in adaptive behavior," *Adapt. Behav.*, vol. 8, no. 2, pp. 129–148, Mar. 2000.
[46] C. Bartolozzi *et al.*, "Embedded neuromorphic vision for humanoid robots," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2011, pp. 129–135.
[47] S. B. Furber *et al.*, "Overview of the SpiNNaker system architecture," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2454–2467, Dec. 2013.
[48] E. Painkras *et al.*, "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, Aug. 2013.
[49] A. P. Davison *et al.*, "PyNN: A common interface for neuronal network simulators," *Front. Neuroinform.*, vol. 2, no. 11, Jan. 2009.
[50] J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie, "Silicon auditory processors as computer peripherals," *IEEE Trans. Neural Netw.*, vol. 4, no. 3, pp. 523–528, May 1993.
[51] A. D. Rast *et al.*, "Transport-independent protocols for universal AER communications," in *Proc. 22nd Int. Conf. Neural Inf. Process. (ICONIP)*, 2015, pp. 675–684.
[52] G. Metta *et al.*, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Netw.*, vol. 23, nos. 8–9, pp. 1125–1134, Oct./Nov. 2010.
[53] M. Peniak, A. Morse, and A. Cangelosi, "Aquila 2.0 software architecture for cognitive robotics," in *Proc. IEEE 3rd Joint Int. Conf. Develop. Learn. Epigenetic Robot. (ICDL)*, Aug. 2013, pp. 1–6.
[54] K. Brohan, K. Gurney, and P. Dudek, "Using reinforcement learning to guide the development of self-organised feature maps for visual orienting," in *Artificial Neural Networks—ICANN* (Lecture Notes in Computer Science), vol. 6353. K. Diamantaras, W. Duch, and L. S. Iliadis, Eds., 2010, pp. 180–189.
[55] F. Galluppi *et al.*, "A real-time, event-driven neuromorphic system for goal-directed attentional selection," in *Proc. 19th Int. Conf. Neural Inf. Process. (ICONIP)*, 2012, pp. 226–233.
[56] S. V. Adams *et al.*, "Towards real-world neurorobotics: Integrated neuromorphic visual attention," in *Proc. 21st Int. Conf. Neural Inf. Process. (ICONIP)*, 2014, pp. 563–570.
[57] A. D. Rast *et al.*, "A location-independent direct link neuromorphic interface," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–8.
[58] T. Binzegger, R. J. Douglas, and K. A. Martin, "A quantitative map of the circuit of cat primary visual cortex," *J. Neurosci.*, vol. 24, no. 39, pp. 8441–8453, Sep. 2004.
[59] R. Bolado-Gomez and K. Gurney, "A biologically plausible embodied model of action discovery," *Front. Neurorobot.*, vol. 7, no. 4, Mar. 2013.
[60] J. C. Anderson, H. Kennedy, and K. A. C. Martin, "Pathways of attention: Synaptic relationships of frontal eye field to V4, lateral intraparietal cortex, and area 46 in macaque monkey," *J. Neurosci.*, vol. 31, no. 30, pp. 10872–10881, Jul. 2011.
[61] J. A. Marchant and C. M. Onyango, "Comparison of a Bayesian classifier with a multilayer feed-forward neural network using the example of plant/weed/soil discrimination," *Comput. Electron. Agricult.*, vol. 39, no. 1, pp. 3–22, Apr. 2003.
[62] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012.
[63] A. Myachykov, R. Ellis, A. Cangelosi, and M. H. Fischer, "Visual and linguistic cues to graspable objects," *Experim. Brain Res.*, vol. 229, no. 4, pp. 545–559, Sep. 2013.
[64] M. H. Degroot, "Changes in utility as information" *Theory Decision*, vol. 17, no. 3, pp. 287–303, 1984.

Alexander D. Rast (M'13) received the Ph.D. degree in computer science from The University of Manchester, Manchester, U.K., in 2011.

He was with the APT Group with The University of Manchester. He is currently a Senior Research Fellow with the University of Southampton, Southampton, U.K. His work has investigated associative language learning on the iCub humanoid robot with biologically realistic spiking neural network models implemented on the SpiNNaker chip. His current research interests include protocols for neuromorphic communications and formal design and implementation methods for spiking neural networks and massively parallel systems in real-world applications.

Samantha V. Adams received the B.Sc. degree in mathematics and physics from The Open University, Milton Keynes, U.K., in 2003, and the M.Res. degree (with Distinction) in computing and the Ph.D. degree in computational neuroscience from the University of Plymouth, Plymouth, U.K., in 2009 and 2013, respectively.

She was a Scientific Software Engineer for many years, and she was a Post-Doctoral Researcher with the University of Plymouth. Her current research interests include biologically inspired computing, encompassing machine learning, AI, and how techniques from these fields can be applied to make smarter applications.

Simon Davidson received the Ph.D. degree from the University of Sheffield, Sheffield, U.K., in 2000, with a focus on neural associative memory.

He spent a decade as a Microprocessor and ASIC Designer with companies including SGS-Thomson, Bristol, U.K., and Hewlett Packard, Grenoble, France and finally with ARC International, Elstree, U.K., where he led the development of the ARC Tangent-A5 processor and codesigned the ARCompact instruction set. He is currently a Research Fellow with The University of Manchester, Manchester, U.K. His current research interests include spiking neural memory and the implementation of massively parallel symbol processing systems on the SpiNNaker platform.

Sergio Davies received the Ph.D. degree in computer science from The University of Manchester, Manchester, U.K.

He was a Research Associate with the APT Group, School of Computer Science, University of Manchester. His current research interests include simulation of spiking neural networks and synaptic plasticity.

Michael Hopkins was a Recording Engineer and a Electroacoustics Designer before running consultancies for 20 years offering experimental design and probabilistic modeling/optimization services to hi-tech engineering companies. He is currently a Research Fellow with the SpiNNaker Group, The University of Manchester, Manchester, U.K., trying to understand how cognitive and other brain functions can emerge from large-scale spiking neural networks, using probability and information theories as the basis of investigation.

Andrew Rowley received the B.Sc. degree in computer science and physics and the Ph.D. degree in computer science from the University of St. Andrews, St Andrews, U.K.

In 2004, he joined The University of Manchester, as a Research Software Engineer, where he has been a Senior Research Software Engineer since 2007. He is currently a Research Fellow and a Senior Research Software Engineer for the Human Brain Project with The University of Manchester, Manchester, U.K, where he is currently a member of the team responsible for the development of the software for SpiNNaker and for the interaction of the SpiNNaker machine with the Human Brain Project Collaboratory.

Alan Barry Stokes received the B.Sc. degree in artificial intelligence and the Ph.D. degree in computer science from The University of Manchester, Manchester, U.K.

He focused on the Human Brain Project in 2013. He is currently a Research Associate and a Research Software Engineer for the Human Brain Project with The University of Manchester, and also a member of the team responsible for the development of the software for SpiNNaker and for the interaction of the SpiNNaker machine with external devices such as retinas and motors. His current research interests include distributed and resilient computing and distributed databases.

Thomas Wennekers received the M.Sc. degree (German diploma) in physics from the University of Düsseldorf, Düsseldorf, Germany, and the Ph.D. degree in computer science from the University of Ulm, Ulm, Germany.

He is currently a Reader of computational neuroscience with the University of Plymouth, Plymouth, U.K. He is an authority on cell assembly models of cortical representations and higher cognitive functions. His current research interests include large-scale, multiple area simulations of biologically realistic spiking-neuron networks, neural computation and learning, and their application to neuromorphic hardware.

Steve Furber (M'99–F'05) received the B.A. degree in mathematics and the Ph.D. degree in aerodynamics from the University of Cambridge, Cambridge, U.K.,

In 1980, he joined Acorn Computers, Cambridge, where he was a Principal Designer of the BBC microcomputer and the ARM 32-bit RISC microprocessor. In 1990, he moved to the ICL Chair at Manchester, U.K., where he leads research into asynchronous and low-power systems and, neural systems engineering, notably the SpiNNaker project. He is the ICL Professor of computer engineering with the School of Computer Science, The University of Manchester, Manchester.

Angelo Cangelosi is currently a Professor of artificial intelligence and cognition and the Director of the Centre for Robotics and Neural Systems with Plymouth University, Plymouth, U.K. His current research interests include language grounding and embodiment in humanoid robots, developmental robotics, human–robot interaction, and on the application of neuromorphic systems for robot learning.

Mr. Cangelosi was the Chair of the IEEE Technical Committee on Autonomous Mental Development from 2012 to 2013. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON AUTONOMOUS DEVELOPMENT, in 2015. He is the Editor (with K. Dautenhahn) of the journal *Interaction Studies*.